Analysis of the Applicability of Reconfigurable Computers in Satellite Telemetry Data Processing

Kenneth B. Winiecki, Jr.; Marco A. Figueiredo; Terry L. Graessle; Parminder S. Ghuman

NASA Goddard Space Flight Center, Greenbelt, Maryland, 20771

ABSTRACT

The advent of reconfigurable computers (RCs) containing field-programmable gate-array (FPGA) ICs presents a potential solution to the problem of processing telemetry data at the high rates required to support the latest remote-sensing satellites. For example, one satellite scheduled for launch in 1999 by NASA's Earth Science Enterprise (ESE) project will generate as much Earth-science telemetry in six months as has been collected in NASA's entire 40-year history. NASA is developing software for large, expensive, conventional parallel-processing computer systems in an attempt to meet the expected processing requirements, but whether or not the resulting performance will be adequate remains unknown. For computationally-intensive, repetitive applications like this, RC technology can provide the critical performance edge.

The Adaptive Scientific Data Processing (ASDP) project at NASA Goddard Space Flight Center (GSFC) has been investigating RC applications in scientific processing systems. ASDP has developed prototype RC solutions which have achieved processing speeds an order of magnitude faster than a conventional high-end computer workstation alone. This paper presents an overview of remote-sensing satellite telemetry, outlines a particular telemetry processing challenge, describes ASDP's application of RC, discusses the results, and analyzes the current and future state of the art.

Keywords: reconfigurable, adaptive, computing, computer, FPGA, NASA, GSFC, ESE, EOS, MODIS, Earth science, remote-sensing, telemetry, processing, acceleration, performance, compute, computation, bound, bottleneck

1. AN OVERVIEW OF NASA REMOTE- SENSING SATELLITE TELEMETRY

A generalization of NASA remote-sensing satellite telemetry is depicted in Figure 1.[1] The satellite (spacecraft) contains one or more instruments, each of which has one or more sets of sensors (detector arrays). A typical detector produces an analog voltage which is converted into a binary digital number. The digital values of all the detectors in an array are periodically and simultaneously sampled, and electronics within the instrument concatenates all the values belonging to a sample and packages them into one or more "packets".[2] A packet is a fixed-length, fixed-format set of binary data consisting of a header followed by a body. The body contains the detector data (or a portion thereof), and the header contains the information necessary to identify the packet from other packets and the data it contains.

All the packets from all the instruments on the spacecraft are collected by a central communications unit which concatenates and distributes them into super-packets called "frames". A frame is a fixed-length, fixed-format set of binary data similar to a packet, but consists, in order, of a sync-mark, header, body, and trailer. The sync-mark is a binary pattern used to distinguish the end of one frame from the beginning of the next; the header contains the information necessary to identify the frame from other frames and the packet data it contains; the body contains one or more packets (or a portion of one); and the trailer contains data that can be used to identify and correct errors which may be introduced into the frame header or body by the transmission process.

The communications unit concatenates the frames to form a data bitstream and sends it to the encoder. The encoder converts (encodes) each data bit into a multiple-bit sequence (symbol), which is more immune to noise than a single data bit, and

Further author information —
K.B.W.: Mail Code 450.L; Tel.: 301-286-3912; Fax: 301-286-0220; Email: kwiniec@vlsi.gsfc.nasa.gov;
Employer: Lockheed Martin Space Mission Systems and Services, Seabrook, MD
M.A.F.: Mail Code 450.L; Tel.: 301-286-7951; Fax: 301-286-0220; Email: marco@fpga.gsfc.nasa.gov;
Employer: Stinger-Ghaffarian Technologies, Inc., Greenbelt, MD
T.L.G.: Mail Code 450.L; Tel.: 301-286-9698; Fax: 301-286-0220; Email: graessle@vlsi.gsfc.nasa.gov;
Employer: Lockheed Martin Space Mission Systems and Services, Seabrook, MD
P.S.G.: Mail Code 564; Tel.: 301-286-5365; Fax: 301-286-0220; Email: ghuman@vlsi.gsfc.nasa.gov;
Employer: NASA (as listed)

passes the symbol bitstream to the modulator. The modulator mixes the symbol bitstream onto the carrier wave, which is then amplified and transmitted.

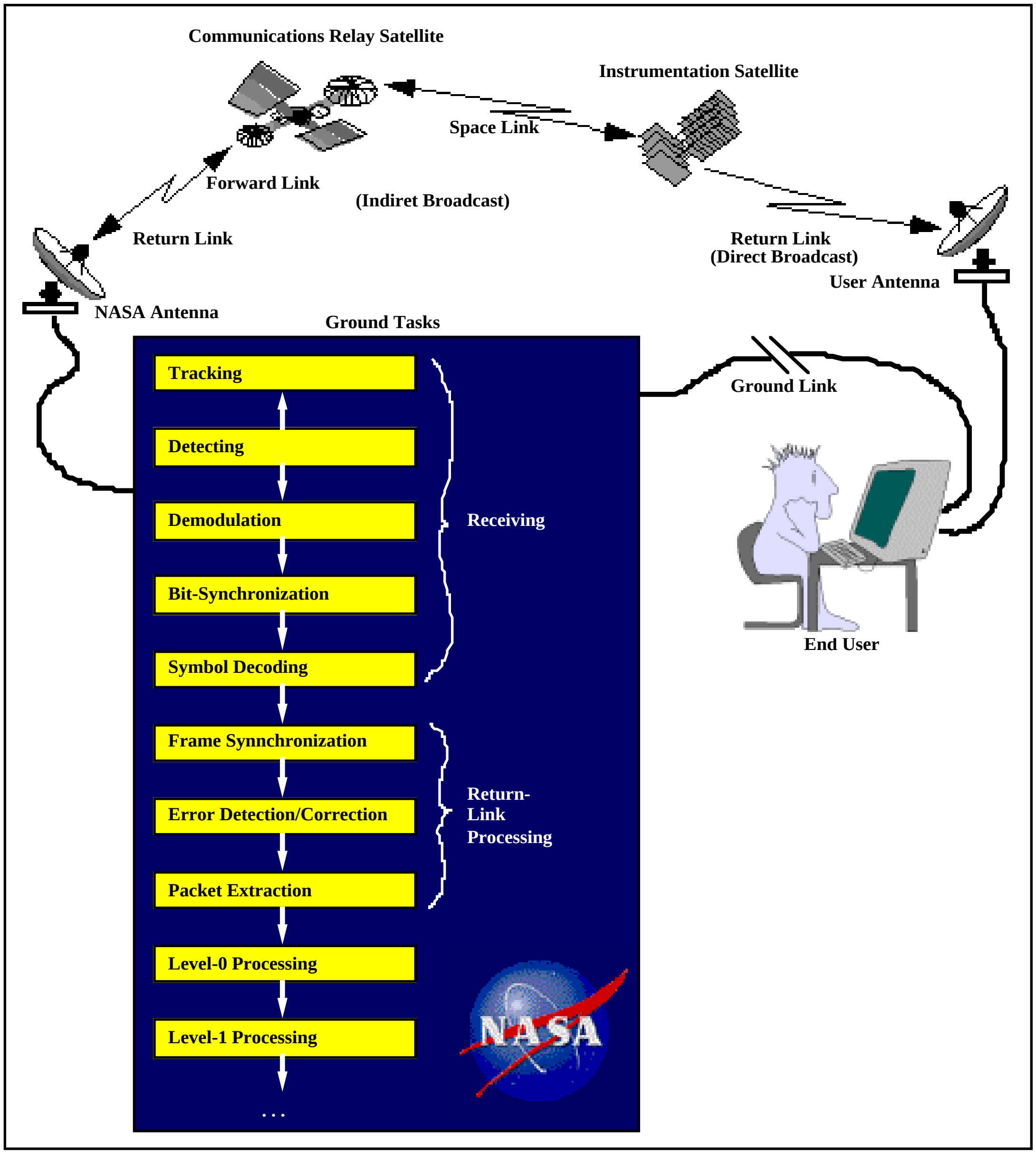


Figure 1: NASA Remote Sensing Satellite Telemetry

The primary transmission mode supported by NASA spacecraft is indirect-broadcast, whereby a higher-bandwidth, lower-power transmission is aimed at the nearest member of NASA's Tracking and Data Relay Satellite System (TDRSS) in a process called "space-link (SL) communication". This communications-relay satellite adds power to the signal and relays it to NASA ground-based communications facilities in a process called "return-link (RL) communication". This is distinguished from "forward-link (FL) communication" which consists of the transmission of spacecraft command and control (C&C) information from the ground to the spacecraft (or to TDRSS). NASA tracks, receives, processes, and distributes the data to the end user in a process called "ground-link (GL) communication".

Another transmission mode supported by many NASA spacecraft is direct-broadcast, whereby a lower-bandwidth, higher-power transmission is aimed at the ground. No SL or GL exists with this mode, so tracking, receiving, and RL processing must be performed by the end user. NASCOM may retain responsibility for C&C, otherwise the end user will also have to provide command-generation and FL transmission capabilities.

The portion of the ground-based communication considered the receiving (RX) domain generally contains tracking, detecting, demodulating, bit-synchronizing, and decoding. Tracking refers to the means to keep the antenna pointed at the spacecraft as it crosses the sky. The antenna contains the detector which picks up the carrier wave which is then amplified and passed to the demodulator. The demodulator eliminates the carrier wave so that only the symbol bitstream remains and sends it to the bit-synchronizer/decoder (BSD). The BSD, knowing how the data bits were encoded into symbol bits, determines the boundaries of the symbols within the symbol bitstream, detects and corrects certain symbol bit errors, and decodes the symbols into data bits forming a data bitstream. At this point the data is considered leaving the RX domain and entering the RL processing domain.

The RL processing domain is generally considered to cover frame-synchronization (FS), error detection and correction (EDC), and packet extraction (service processing or SP). The data bitstream from the receiver enters the FS which searches for the sync-pattern, identifies the boundaries of the frames, keeps track of mis-sized frames, aligns the data bits into bytes, and passes the bytestream and quality information to the EDC. The EDC uses the information in the frame trailer to locate and repair data bit errors (up to some maximum number) in the frame header or body, and passes the corrected bytestream and quality information to the SP. The SP uses the information in the frame header and the packet headers to extract the packets and/or packet pieces, reassemble packets from pieces, generate filler for missing packet data, index packets, and store them in a single file. At this point the data is considered leaving RL processing and entering Level-Zero (L0 or LZ) processing.

L0 is defined as taking RL output and reconstructing as closely as possible the original packet stream as generated by each instrument (of interest) on board the spacecraft. It involves sorting packets, reordering, discarding duplicates, generating filler, creating files, and record-keeping. The packet file generated by RL processing may contain packets generated by different instruments, or different detector arrays within an instrument, all mixed together. It may contain duplicate packets due to an on-board situation or ground command that caused it to re-transmit a portion of the data. It may be missing packets due to unrecoverable errors or temporary losses of signal. The packets may appear out-of sequence if they were transmitted from an on-board storage unit (recorder) that accumulated packets or frames while the spacecraft was unable to transmit normally. L0 processing stores complete, sorted, ordered packet streams in separate files, and stores indices and quality information in other files, all of which are passed to Level-One (L1).

Generalizations cease to be possible at L1 because it is highly application-specific. L1 basically covers reconstructing as closely as possible from the L0 output the datasets produced by the detector arrays in their native granularity before they were packetized by the instrument, and then may include various forms of data adjustment and interpolation necessary to get the data into the final state required to generate a user-understandable form (such as an image). Actually generating and displaying the image, archiving and retrieving images, and combining and analyzing images are the objects of Level-Two (L2) and higher levels of processing. Level designations are then basically used to denote processing steps that generate data products of some more general use; level designations have been known to approach 50.

2. THE PROBLEM OF LEVEL-ONE PROCESSING OF MODIS DATA

It should be evident from the preceding discussion of remote-sensing satellite telemetry that the initial stages of ground-based processing, from receiving through the reconstruction subtask of L1, are highly I/O-bound and computationally trivial. As such, they are not particularly appropriate for reconfigurable computing technology. Where data rates are low (less than 1 Mb/s), all the processing tasks can be performed in software running on a series of modern mid-range computer workstations. Where they are higher (up to 400 Mb/s), NASA GSFC has developed application-specific integrated circuits (ASICs) to perform all of the RX (pending) and RL tasks, multi-CPU systems and software to perform L0 processing, PCI and VME cards that apply the ASICs, and complete prototype systems, many of which have been deployed around the world. Furthermore, all of these advances and more are (or soon will be) obtainable by industry free of charge through NASA's Technology Transfer Program.[3]

It should also be clear that L1 after the reconstruction subtask, and many higher processing levels, are compute-bound, and thus are excellent candidates for reconfigurable computing. This was demonstrated in 1997 by ASDP in a study for the ESE (formerly Mission to Planet Earth, MTPE) Earth Observing System (EOS) Data and Information System (EOSDIS).[4] The first of the EOS spacecraft, the Ante Meridian #1 (EOS-AM1),[5] is currently awaiting launch in 1999. It contains five instruments which together generate one terabyte (TB, 1024 GB) of telemetry per day. In six months it will have generated 182 TB, more than the sum total of Earth-science data collected by NASA up to this point (approximately 125 TB).[6] It is expected to run for at least six years, generating a total of more than two petabytes (PB, 1024 TB). The EOS-AM1 spacecraft is shown in Figure 2.

The L1 processing software for just one of the instruments, the Moderate-Resolution Imaging Spectroradiometer (MODIS),[7] is being developed in the C programming language by the MODIS Characterization Support Team (MCST)[8] to run on a high-end multiprocessing server, the SGI Challenge,[9] yet at the time, an issue remained as to whether or not it would be able to keep up with the data. The MODIS instrument views the Earth in 36 different bands of the electromagnetic radiation spectrum (from far-infrared through visible) accomplished with 36 rows of detectors. 29 bands have 1000 m resolution, five have 500 m, and the remaining two have 250 m. A detector row can see 10 km at NADIR (straight down), which means that a 1000 m row is 10 detectors long, 500 m is 20, and 250 m is 40. That translates to a total of 470 detectors, which can be thought of as pixels. The detectors are periodically sampled with 12 bits of precision, so a sample can be considered to be a multispectral set of rows of 12-bit pixel data. Using mirrors, an image of the Earth is continually rotated over the detectors, which is conceptually equivalent to rotating the detectors over the image of the Earth. The detectors are sampled at a rate proportional to their size: the 1000 m detectors are sampled 1354 times, the 500 m 2708 times, and the 250 m 5416 times, during each time the image of the Earth is moving across the detectors, generating a set of 36 swath images called a scan which are 2200 km long; this is depicted in Figure 2.

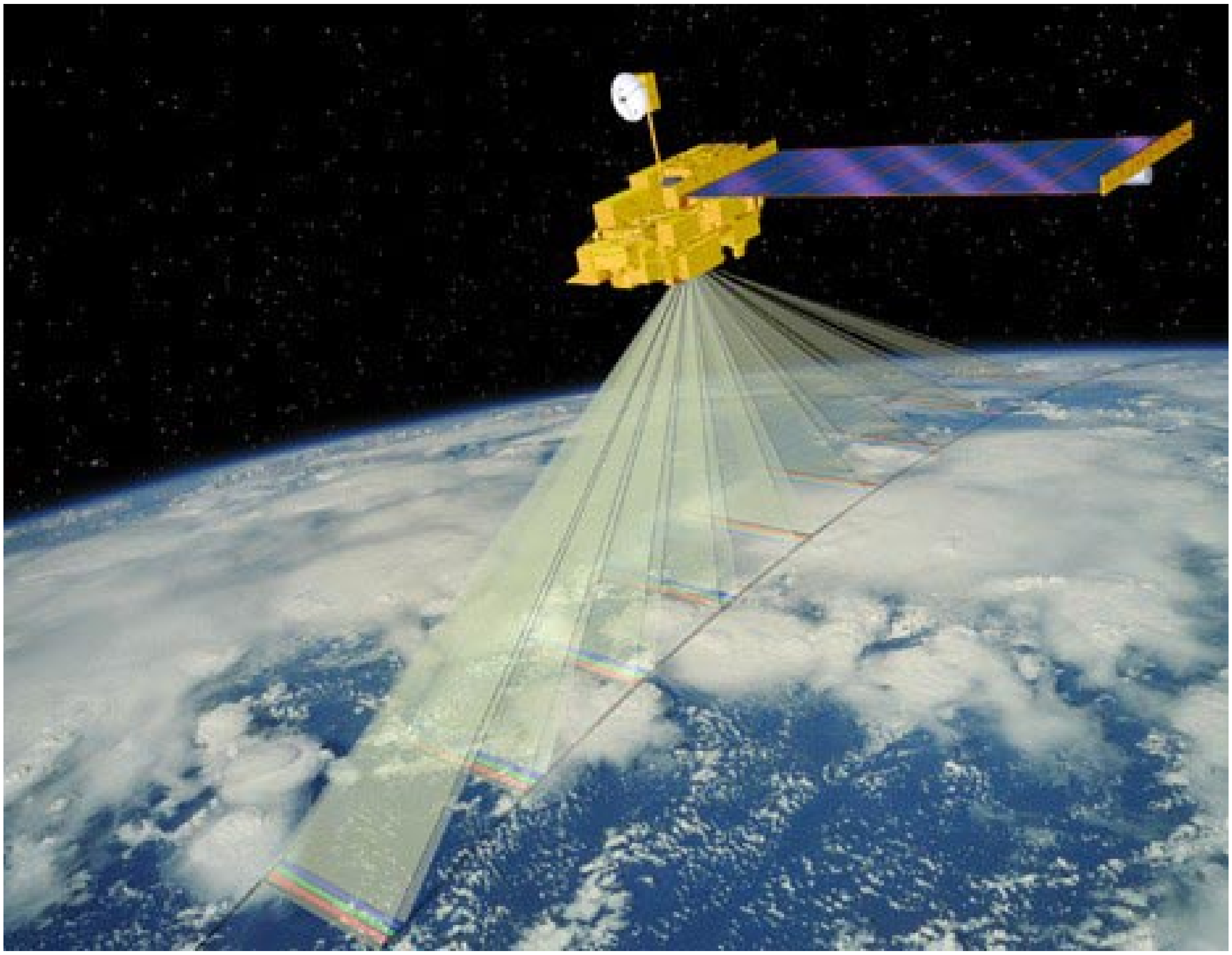


Figure 2: NASA EOS-AM1 Spacecraft with MODIS Scanning Operation

Summing the twenty-nine 1354 x 10 -pixel images, five 2708 x 20s, and two 5416 x 40s in a scan results in 1,096,740 pixels. At 12 bits per pixel, a scan contains 1.6 MB of Earth-view data, plus approximately 200kB of calibration and engineering data. Note that due to the curvature of the earth, the swath is 10 km wide only in the center, and spreads out to 25 km wide at the scan boundaries. The Earth is scanned once every 1.477 seconds, and exactly five minutes worth of data (203 scans, almost 100 MB) has been termed a "data granule". The entire surface of the earth is covered in about every two days.

The MODIS L1 software is divided into three modules: L1A, Geolocation, and L1B. L1A contains the aforementioned reconstruction of detector array datasets (scans and granules). Geolocation is accurately locating each pixel in all three dimensions with respect to the Earth. L1B contains procedures for calibrating (normalizing) the pixel data for the source detector's inherent variation and degradation against a number of known test radiation sources (data from which accompanies every scan), as well as routines for pixel aggregation (creating equivalent lower-resolution scans out of higher-resolution scans). Block diagrams of the MODIS L1 and L1B task organization appear in Figure 3.

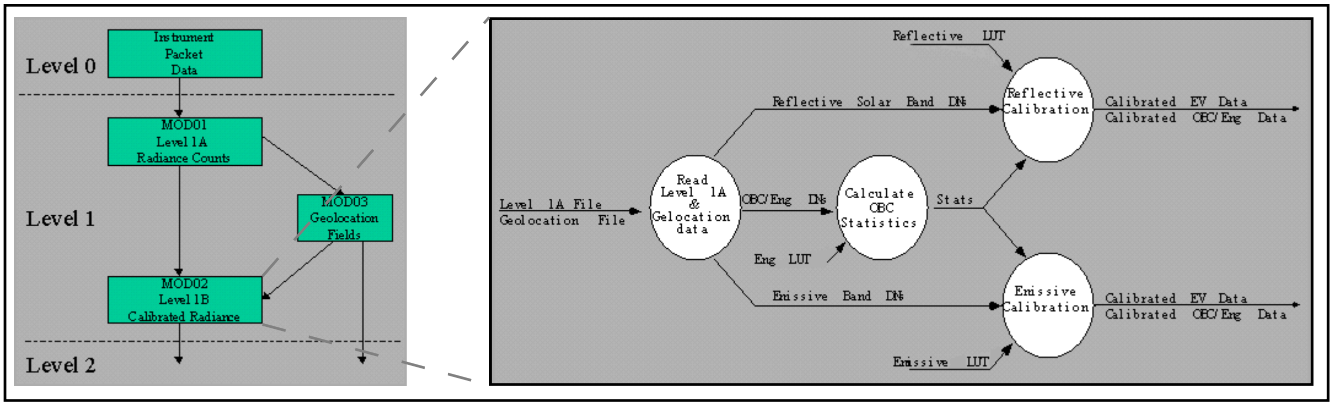


Figure 3: MODIS Level 1 Processing Tasks

3. AN ASDP APPLICATION OF RECONFIGURABLE COMPUTING

To demonstrate the potential value of reconfigurable computing in accelerating the MODIS L1 software, ASDP targeted the first compute-bound L1 module, L1B. To minimize the effort of porting the code from the MCST environment to the ASDP environment, the least expensive SGI computer with a fast bus was selected to be the host: the SGI Origin 200 Server with PCI, shown here in Figure 4.

Upon analyzing L1B, ASDP determined that the Reflective Calibration subroutine, which calibrates all the Earth-view data in a granule from each detector against the same detector's Sun-view data (its output from viewing a reflected, filtered solar image), was the largest computational bottleneck, consuming 33% of the total processing time. For the reconfigurable computer, ASDP selected a single Annapolis Microsystems WildForce PCI Reconfigurable Computer, shown here in Figure 4.[10]

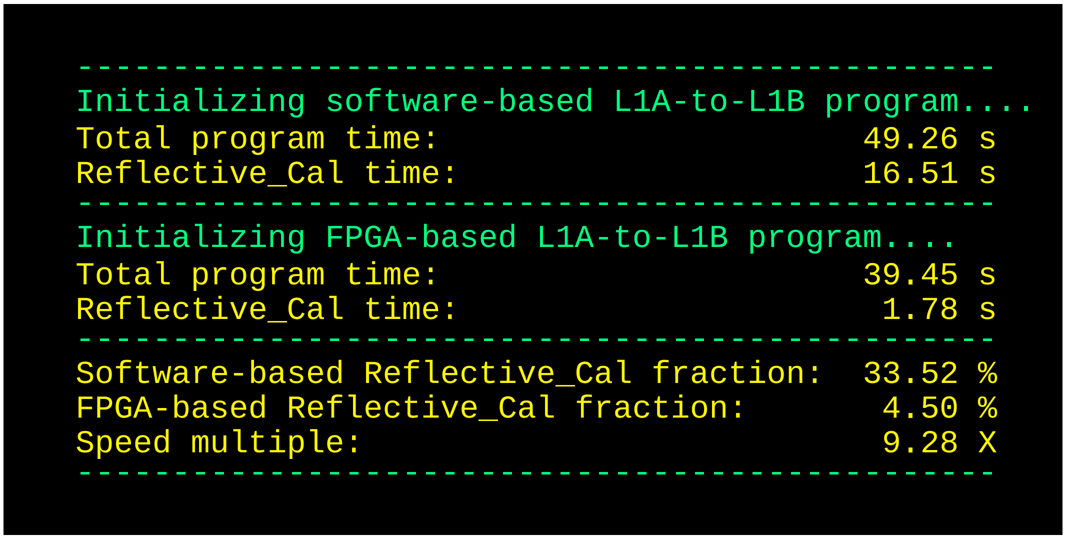
Figure 4: SGI Origin 200 Server and Annapolis Microsystems WildForce PCI Reconfigurable Computer

The WildForce model available at the time contained 180k gates (or 13k Xilinx "configurable logic blocks"), 10 MB SRAM, and 3 FIFO buffers. In particular, it had five Xilinx FPGAs -- one 4020EX with one FIFO buffer and 2 MB SRAM, and four 4036EX devices connected in a chain with input and output FIFO buffers and 2 MB SRAM per device. Additionally, all five FPGAs were connected by a configurable crossbar switch, and all FIFO buffers and memory banks were directly accessible through the PCI bus.

ASDP identified and implemented the most compute-intensive portion of the (floating-point) Reflective Calibration subroutine in fixed-point on the WildForce. The implementation was super-pipelined, and produced one result every 30 MHz clock cycle after the pipeline was filled. The MODIS code was modified the minimum amount necessary to operate the hardware and eliminate the code for which hardware had been developed, the two versions of L1B were executed side-by-side, and timing results were recorded. For details about the development, please refer to the "Adaptive Scientific Data Processing -- Fall 1997 Report".[4]

4. RESULTS

The input data granule used by MCST and supplied to ASDP contained only ten scans instead of the 203 that form a complete granule, so the test ran relatively quickly. A screen-dump of the combined (modified) L1B program output is displayed in Figure 5.



```
------------------------------------------------
Initializing software-based L1A-to-L1B program....
Total program time:                      49.26 s
Reflective_Cal time:                     16.51 s
------------------------------------------------
Initializing FPGA-based L1A-to-L1B program....
Total program time:                      39.45 s
Reflective_Cal time:                      1.78 s
------------------------------------------------
Software-based Reflective_Cal fraction:  33.52 %
FPGA-based Reflective_Cal fraction:       4.50 %
Speed multiple:                           9.28 X
------------------------------------------------
```

Figure 5: Results

The most important observation is that the Reflective Calibration processing time dropped 89.2% (14.73 s), from 16.51 s to 1.78 s, achieving 9.28 times the original speed.

Another important observation is that the fraction of the total processing time devoted to Reflective Calibration was reduced 86.6%, from 33.52% to 4.50%, indicating that Reflective Calibration was no longer a bottleneck in L1B processing.

5. THE STATE OF THE ART

The field of reconfigurable computing is still in its infancy. The enabling technology was born in 1985 with the introduction of small SRAM-based FPGAs from Xilinx, Inc. These devices offered the new advantage of hot, in-circuit logic reconfigurability. In the years following, FPGA architecture has been steadily improved, and presently, devices are available which run at clock speeds over 80 MHz and offer over 100k gates. By next year (1999) this number is expected to rise to over 1 million, and this trend can continue as long as semiconductor fabrication techniques continue to evolve.

The two most prevalent applications for FPGAs are ASIC prototyping and low-volume embedded systems; reconfigurable computing currently represents only a small segment of the FPGA market and is hardly considered a significant source of revenue by device manufacturers. However, reconfigurable computing has exhibited sustained growth and is widely recognized as the area with the greatest potential for both future growth and for establishing FPGAs in the marketplace as mainstream computing elements. The work of the ASDP project at NASA GSFC visibly supports this conclusion, as does work elsewhere on other problems that have demonstrated computational accelerations of several orders of magnitude.

The successful development of reconfigurable computing applications currently requires designers with in-depth knowledge of both hardware and software engineering issues, i.e., computer engineers. Algorithms must be painstakingly analyzed to determine their fitness for implementation in FPGAs and on the various available system architectures, the architectures must be researched and the system(s) selected. Sufficiently large algorithms must be carefully partitioned between hardware and software, and then the hardware partition must itself be partitioned among the individual FPGAs in the system (and the individual systems if applicable). Care must be exercised to ensure that the input data required by each FPGA (whether from software or from another FPGA) is properly generated and has a viable datapath.

The size or nature of some applications warrants the use of dynamic (run-time) reconfiguration. Various granularities of reconfigurability are available, from reconfiguring just portions of a single FPGA to reconfiguring an entire system of FPGAs. Implementing such a system requires the design of multiple logic configurations to occupy the same space as well as software to support reconfiguring on-the-fly. It also requires that the time required to perform a reconfiguration (up to several milliseconds) and any consequential on-board memory required to buffer data during the reconfiguration downtime be accountedfor.

As previously mentioned, only compute-bound processing tasks are appropriate for the application of reconfigurable computing -- I/O-bound tasks are not. This is as expected when comparing the aggregate throughput of FPGAs, which are massively parallel logic gates, against traditional address and data busses used to transfer data into and out of them, such as PCI. This becomes even more evident when the effects of operating system overhead on bus utilization are considered. Therefore, determining the I/O and computation profiles of a task before beginning the application of reconfigurable computing is essential.

If reconfigurable computing is to move into the mainstream of computing, much more advanced development tools will be required to simplify the design process. Current tools are attempting to address top-down reconfigurable computing application design, but they are still immature. An ideal tool would allow all of the system parameters to be specified using a high-level symbolic or graphical interface. Such a tool would give the scientist the means to enter the specification for their application directly from its mathematics, and would facilitate for the computer engineer the partitioning of the task between the FPGAs, systems, and software, as well as the implementation of the design. It would also have the ability to be easily upgraded to take advantage of the latest FPGA technology and changing system architectures, to reduce the need to continually procure new tools.

In addition to better tools, the architecture of FPGAs will have to evolve to better facilitate the unique requirements of the reconfigurable computing paradigm. This means supporting partial-FPGA reconfiguration down to the gate or logic cell level, reducing the dynamic reconfiguration cycle time down into the microsecond range, and increasing clock speeds. With the current limitations, dynamic reconfigurability is not a viable option for any but a relatively small number of applications, although the room for improvement is great.

And finally, forward-compatibility needs to be designed into FPGA system architectures and products. Currently, FPGA density increases and architecture changes occur rapidly enough that system obsolescence is a major problem. This forces system vendors to continually redesign their products, and this in turn severely detracts from the perceived benefit of reconfigurable computers. For the adoption of reconfigurable computing to become widespread, upgrading FPGA systems to keep up with the latest FPGA offerings must become less physically, logistically, and financially painful.

The sole justification for utilizing today's reconfigurable computers is the potential to attain orders-of-magnitude processing performance gains over current microprocessor-based systems. The primary obstacle to overcome in the realization of this potential is the difficulty of developing the application. Several efforts are underway to address the limitations put forth, supported by the Defense Advanced Research Projects Agency (DARPA)'s Adaptive Computing Systems study.[11] The ASDP group at GSFC is collaborating with such efforts by providing an interface between the research community and those working on the reconfigurable computing challenges facing NASA.

6. CONCLUSION

Today's reconfigurable computing technology has been shown to provide an order of magnitude improvement in processing speed to a particular remote sensing data processing task, MODIS Level 1-B Reflective Calibration. The ASDP group at NASA GSFC has thus demonstrated the enormous potential of reconfigurable computing to appreciably speed up compute-bound, repetitive processing tasks in general such as those found at NASA Level 1 and higher.

The obstacles to the widespread adoption of reconfigurable computing have included FPGA logic density and architecture, FPGA system lifetime and upgradability, and development tool software. Since reconfigurable computing is recognized as the FPGA application with the greatest growth potential, reducing or eliminating the obstacles should significantly encourage market expansion. Work is underway to address these issues, but there remains room for more.

7. ACKNOWLEDGMENTS

This effort was supported by the National Aeronautics and Space Administration (NASA).

8. REFERENCES

1. Ramapriyan, H.K., "An Overview of EOSDIS", http://spsosun.gsfc.nasa.gov/NewEOSDIS_Over.html , July 1998.

2. "Advanced Orbiting Systems, Networks, and Data Links", Consultative Committee for Space Data Systems, CCSDS 701.0-B-2, Blue Book, Issue 2, November 1992, Annapolis, MD.

3. "Microelectronic Systems Branch", http://vlsi22.gsfc.nasa.gov/public/ , Nicholas Speciale, 1997.

4. Figueiredo, M., T. Graessle, K. Winiecki, D. Greenspan, "Adaptive Scientific Data Processing -- Fall 1997 Report", http://fpga.gsfc.nasa.gov/asdp/fall97rep/fall97rep.htm , September 1997

5. Scolese, C., "EOS-AM Home Page", http://eos-am.gsfc.nasa.gov/index.html , July 1998.

6. Obenschain, C., "EOS Data and Information System (EOSDIS), Status and Plan", slide presentation, January 16 1998.

7. Conboy, B., "MODIS", http://ltpwww.gsfc.nasa.gov/MODIS/ , July 1998.

8. Guenther, B., "MODIS Characterization Support Team", http://ltpwww.gsfc.nasa.gov/MODIS/MCST/Home.html , March 1998

9. "Silicon Graphics Inc.", http://www.sgi.com/ , August 1998.

10. "Annapolis Micro Systems, Inc.", http://www.annapmicro.com/ , August 1998.

11. DARPA ITO Adaptive Computing Systems study, http://www.darpa.mil/ito/research/index.html